INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A randomized Quasi-Monte Carlo method for the simulation of product-form loss networks

Bruno Tuffin

N ° 3671

April 27, 1999

THÈME 1

ISSN 0249-6399

A randomized Quasi-Monte Carlo method for the simulation of product-form loss networks

Bruno Tuffin *

Thème 1 — Réseaux et systèmes
Projet Armor

Rapport de recherche n3671 — April 27, 1999 — 18 pages

Abstract: In this paper, we use low discrepancy sequences as a variance reduction tool for the Monte Carlo simulation of loss models in broadband telecommunication networks. We have already applied this technique to product-form multi-class queuing networks and to a particular loss model representing a cellular system with dynamic resource sharing. We generalize here this last result to a broader class of loss models and we give some enlightenments and precise details on the behavior of the method.

Key-words: Telecommunications, Monte Carlo, Variance Reduction, Low discrepancy sequences.

(Résumé : tsvp)

* btuffin@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : 02 99 84 71 00 - International : +33 2 99 84 71 00
Télécopie : 02 99 84 71 71 - International : +33 2 99 84 71 71

Une méthode de quasi-Monte Carlo randomisée pour la simulation des réseaux à perte à forme produit

Résumé : Dans ce rapport, nous utilisons les suites à discrépance faible comme outil de réduction de la variance pour la simulation des modèles à perte dans les réseaux de télécommunication à large bande. Nous avons déjà appliqué cette technique aux réseaux de files d'attente multi-classes à forme produit et à un modèle à perte particulier représentant un système cellulaire avec partage dynamique des ressources. Nous généralisons ici ce dernier résultat à une classe plus large de modèles à perte et nous donnons quelques éclaircissements et détails précis sur le comportement de la méthode.

Mots-clé : Télécommunications, Monte Carlo, Réduction de la variance, Suites à discrépance faible.

1 Introduction

The interest in loss models has increased in the last decades, due to the broad and growing prominent class of systems they represent. A system from which resources are available is called a *loss system* if, when an arrival customer does not found a sufficient amount of required resources, this customer is lost. The first model of this kind is the Erlang loss model which has resulted in the well-known Erlang loss formula (Ross 1995, Chapter 1). A lot of work has been done ever since to generalize this expression.

The loss model we consider can represent circuit-switched telephone networks, single service telephone networks with fixed routing, ATM networks, as well as the stochastic knapsack problem. Using some minor modifications, it is also possible to modelize other systems such as, for instance, ATM networks with bidirectional links or ATM networks with route separation or with multiplexing across routes.

In the sequel, the arrival calls in the model are supposed to be exponentially distributed, which is not necessarily the case for holding times, so that the stationary probabilities have a product-form solution known apart from a normalization constant. Performance measures can be expressed as simple functions of those normalization constants, whose computation is then an important challenge. When the state space is large, a direct computation is intractable but there exists some efficient methods using for instance convolution or fast Fourier transforms (Kobayashi and Mark 1997, Ross 1995). Nevertheless, these combinatorial algorithms can not be used for all topologies, so that we have sometimes to use Monte Carlo methods as in (Ross 1995, Ross and Wang 1992), where Monte Carlo methods using importance sampling to reduce the estimator's variance have been applied.

In the present paper, we improve these last results by using randomized quasi-Monte Carlo methods developed in (Cranley and Patterson 1976, Tuffin 1996a, Tuffin 1997a, Tuffin 1997b, Tuffin 1998). Such techniques use low discrepancy sequences (Niederreiter 1992), which have the property to be quickly uniformly distributed over the integration space, as a variance reduction technique. This can lead to great improvements by reducing the confidence interval width with respect to a standard Monte Carlo method. This technique has already been applied in (Ross 1995) to product-form multi-class queuing

networks (as an improvement of the work done in (Ross and Wang 1997) and in (Tuffin 1996b) to a particular loss model which represents a cellular system with dynamic resource sharing (Fleming et al 1995). We consider here the generalized class of loss models studied in (Ross 1995) and apply our technique to these models. It is a more general model than the one described in (Tuffin 1996b), which, using trivial adjustments, fits the same kind of framework.

The paper is organized as follows. In Section 2 we present the model on which we will work and in Section 3 we recall the Monte Carlo method of (Ross 1995, Ross and Wang 1992). Next, we describe the randomized quasi-Monte Carlo method in Section 4 and we give numerical experimentations in Section 5. Finally, we describe in Section 6 an integral representation of normalization constants developed in (Kobayashi and Mark 1997). Simulation using this integral form seems to be a promising direction for further work, especially if randomized quasi-Monte Carlo methods are used. We explain then what are the current limitations of this kind of simulation for the present form.

2 Model

The model we use here is more detailed in (Ross 1995) and represents a wide range of telecommunication applications. We consider a system consisting in J links with a capacity of C_j bandwidth units for link j. There are also K classes of calls. Each call class k ($k = 1, \cdots, K$) is characterized by

- Independent Poisson arrival processes with rate λ_k.
- Independent holding times (not necessarily Poisson processes) with mean $1/\mu_k$.
- A route $R_k \subset \{1, \cdots, J\}$ which is a subset of links required for example to connect two nodes.
- The b_k bandwidth units required in each link $j \in R_k$ for a call. In the case of ATM networks with peak rate admission, b_k denotes the peak rate of a class k service ($k = 1, \cdots, K$). In the case of statistical multiplexing, it denotes the effective bandwidth.

Suppose that a call of class k arrives. If there is a link $j \in R_k$ such that more than $C_j - b_k$ bandwidth units are already used, then the call is lost. Otherwise, the call is admitted and b_k is deduced of the available capacity of each link $j \in R_k$ until the service is completed.

As in (Ross 1995), define $\mathcal{K}_j$ as the set of classes using link j, i.e., $\mathcal{K}_j = \{k \in \{1, \cdots, K\} \ : \ j \in R_k\}$. A state of the system is given by the vector

$$\boldsymbol{n} = (n_1, \cdots, n_K)$$

where n_k is the number of calls of class k $(k = 1, \cdots, K)$ in the system. The state space is then

$$\mathcal{S} = \left\{ \boldsymbol{n} \in \mathbb{N}^K : \sum_{l \in \mathcal{K}_j} b_l n_l \leq C_j \ \ \forall 1 \leq j \leq J \right\}.$$

Moreover, the set $\mathcal{S}_k$ of states for which a call of class k $(k = 1, \cdots, K)$ is admissible is given by

$$\mathcal{S}_k = \left\{ \boldsymbol{n} \in \mathcal{S} : \sum_{l \in \mathcal{K}_j} b_l n_l \leq C_j - b_k \ \forall j \in R_k \right\}.$$

Denote by $\boldsymbol{X} = (X_1, \cdots, X_K)$ the equilibrium random vector such that X_k $(k = 1, \cdots, K)$ is the steady state number of class-k calls. Let $\rho_k = \lambda_k / \mu_k$ for $k = 1, \cdots, K$ be the traffic intensity of class k. Following (Ross 1995), $\boldsymbol{X}$ has the following distribution

$$\mathbb{P}(\boldsymbol{X} = \boldsymbol{n}) = \frac{1}{G} \prod_{k=1}^{K} \frac{\rho_k^{n_k}}{n_k!}$$

$\forall \boldsymbol{n} \in \mathcal{S}$ where

$$G = \sum_{\boldsymbol{n} \in \mathcal{S}} \prod_{k=1}^{K} \frac{\rho_k^{n_k}}{n_k!}$$

is the normalization constant.

For link $j \in \{1, \cdots, J\}$, let $U_j = \sum_{k \in \mathcal{K}_j} b_k X_k$ be the equilibrium amount of bandwidth used. An important measure on this model is the asymptotic blocking probability for a class-k call

$$B_k = 1 - \mathbb{P}(U_j \leq C_j - b_k \ \forall j \in R_k) = 1 - \frac{G_k}{G}$$

where $G_k = \sum_{\boldsymbol{n} \in \mathcal{S}_k} \prod_{k=1}^{K} \rho_k^{n_k}/n_k!$ is the normalization constant for the same system but with a bandwidth capacity $C_j - b_k$ instead of C_j in each link $j \in R_k$, i.e., using state space $\mathcal{S}_k$ instead of $\mathcal{S}$ (Ross 1995, page 160). Note that these constants allow to compute many other performance measures, so that the knowledge of their values is a challenging point.

There exist efficient combinatorial algorithms to compute the normalization constants and then the loss probabilities, but these methods do not work for all topologies (i.e., shape of R_k $\forall k$) (Ross 1995). In fact, computing these values is a NP-complete problem, so that we use simulation techniques.

As a last remark, note that this model is very general; with very simple modifications on the state space $\mathcal{S}$, it can also represent (Ross 1995)

- an ATM network with dynamic service/dynamic-route separation;
- an ATM network with static or dynamic-service separation/multiplexing across routes;
- a cellular system with dynamic resource sharing as described in (Fleming et al 1994).

3 Monte Carlo simulation

As the normalization constants are unknown, it is impossible to simulate directly the random vector $\boldsymbol{X}$, so we simulate a random vector $\boldsymbol{Y}$ with a distribution the closest to $\boldsymbol{X}$. By this, we do what is called in Monte Carlo theory *importance sampling* (Fishman 1997). Let us explain how it is implemented in (Ross 1995, Ross and Wang 1992).

For $1 \leq k \leq K$, let $N_k = \min\{C_j/b_k : j \in R_k\}$ be the maximal possible number of class-k calls in the system and Y_k be a truncated Poisson random variable with parameter γ_k and values in $\{0, \cdots, N_k\}$. To facilitate the carrying out, we suppose that the Y_k $(k = 1, \cdots, K)$ are independent. Define also $\boldsymbol{Y} = (Y_1, \cdots, Y_K)$ with values in $\tilde{\mathcal{S}}$ with

$$\tilde{\mathcal{S}} = \{0, \cdots, N_1\} \times \cdots \times \{0, \cdots, N_K\},$$

and note p its probability distribution, which is therefore a product of truncated Poisson laws. Then

$$G = \sum_{\boldsymbol{n} \in \tilde{\mathcal{S}}} f(\boldsymbol{n}) = \mathbb{E}_p \left(\frac{f(\boldsymbol{Y})}{p(\boldsymbol{Y})} \right) \tag{1}$$

where $f(\boldsymbol{n}) = \prod_{k=1}^{K} \frac{\rho_k^{n_k}}{n_k!} \mathbb{1}(\boldsymbol{n} \in \mathcal{S})$ and $\mathbb{E}_p(\cdot)$ is the expectation with respect to probability measure p.

Using I independent vectors $\boldsymbol{Y}^{(i)}$ with the same law as $\boldsymbol{Y}$, estimators of G and B_k $(k = 1, \cdots, K)$ are respectively

$$\bar{\Phi}_I = \frac{1}{I} \sum_{i=1}^{I} \Phi(\boldsymbol{Y}^{(i)})$$

and

$$\bar{\Gamma}_I^{(k)} = \frac{\sum_{i=1}^{I} \Phi^{(k)}(\boldsymbol{Y}^{(i)})}{\sum_{i=1}^{I} \Phi(\boldsymbol{Y}^{(i)})}$$

with $\Phi = f/p$ and $\Phi^{(k)}(\boldsymbol{Y}^{(i)}) = \Phi(\boldsymbol{Y}^{(i)}) \mathbb{1}(\boldsymbol{Y}^{(i)} \in \mathcal{S}_k)$. Using the Central Limit Theorem as $I \to \infty$, we obtain a confidence interval at risk 5% for B_k $(k = 1, \cdots, K)$

$$\left[\bar{\Gamma}_I^{(k)} - 1.96 \frac{s_I^{(k)}}{\bar{\Phi}_I^{(k)} \sqrt{I}}, \bar{\Gamma}_I^{(k)} + 1.96 \frac{s_I^{(k)}}{\bar{\Phi}_I^{(k)} \sqrt{I}} \right]$$

where

$$(s_I^{(k)})^2 = \sigma_I^2(\Phi^{(k)}) - 2\bar{\Gamma}_I^{(k)} Cov(\Phi, \Phi^{(1)}) + (\bar{\Gamma}_I^{(k)})^2 \sigma_I^2(\Phi)$$

can easily be estimated (Fishman 1997).

All the problem is reduced now to the choice of parameters γ_k $(k = 1, \cdots, K)$. The heuristic developed in (Ross 1995, Ross and Wang 1992) is to take

$$\gamma_k = (1 + .15(1 - z))^{b_k} \rho_k \text{ for } k = 1, \cdots, K$$

where $z = \max_{1 \le j \le J} (\sum_{k \in \mathcal{K}_j} b_k \rho_k)/C_j$ is assumed not to be greater than one.

4 Simulation using Randomized quasi Monte Carlo methods

4.1 Quasi-Monte Carlo methods

Quasi-Monte Carlo methods are devoted to compute the integral of a function g on the domain $[0,1]^s$, say, to compute $\int_{[0,1]^s} g(x)dx$. We describe here how these techniques operate and we will see in sub-section 4.3 how to use them to our performance evaluation problem which is to work out a summation instead of an integration, over a different kind of space.

Quasi-Monte Carlo methods (Niederreiter 1992) are deterministic analogs of Monte Carlo ones and use *low discrepancy sequences* $(\xi^{(i)})_{i\in\mathbb{N}}$ which have the property of having a fast uniform distribution. The approximate value of $\int_{[0,1]^s} g(x)dx$ is then $\frac{1}{I}\sum_{i=1}^{I} g(\xi^{(i)})$ and an error bound is given by the Koksma-Hlawka Theorem (Niederreiter 1992):

$$\left|\frac{1}{I}\sum_{i=1}^{I} g(\xi^{(i)}) - \int_{[0,1]^s} g(x)dx\right| \leq V(g)D_I^*\left((\xi^{(i)})_{i\in\mathbb{N}}\right)$$

where $V(g)$ is the variation of g and

$$D_I^*((\xi^{(i)})_{i\in\mathbb{N}}) = \sup_{x\in(0,1]^s}\left|\frac{1}{I}\sum_{i=1}^{I}\mathbb{1}_{[0,x)}(\xi^{(i)}) - \prod_{j=1}^{s} x_j\right|$$

is the discrepancy of the I first elements of the sequence $(\xi^{(i)})_{i\in\mathbb{N}}$. The convergence speed is then given by the decreasing speed of the discrepancy. For low discrepancy sequences, this speed is $O(I^{-1}(\ln I)^s)$, which is conjectured to be the best possible (Niederreiter 1992, Tuffin 1997a). Consequently, if $V(g)$ is finite, the Koksma-Hlawka bound leads to an error in $O(I^{-1}(\ln I)^s)$, faster than the one of Monte Carlo methods which is in $O(I^{-1/2})$.

But the problem with these methods is that, even if the convergence is fast, the error is unknown in practice because usual error bounds like the Koksma-Hlawka one depend on unknown or extremely large quantities. For example, this is the case of the variation, which is in general unknown *and*

large, and of the discrepancy for which the only valuable known behavior is the asymptotic one. Thus, we combine Monte Carlo and quasi-Monte Carlo methods to benefit of the advantages of each of them: fast convergence speed for quasi-Monte Carlo and error estimate (by means of a confidence interval) for Monte Carlo.

4.2 Randomized-quasi-Monte Carlo methods

We consider here I random variables

$$Z^{(i)} = \frac{1}{L}\sum_{l=1}^{L} g(\{U^{(i)} + \xi^{(l)}\}),$$

$i = 1, \cdots, I$, where the $U^{(i)}$ are i.i.d. random variables uniformly distributed over $[0,1]^s$ and $\{\cdot\}$ denotes the fractional part of each coordinate of a vector of $\mathbb{R}^s$. With respect to quasi-Monte Carlo methods, a random part is then inserted in $Z^{(i)}$ by adding the same random variable to each point of the low discrepancy sequences in the quasi-Monte Carlo estimation. We hope that the uniform distribution of $(\xi^{(i)})_{i\in\mathbb{N}}$ will reduce the variance of $Z^{(i)}$ with respect to the one of the average of L i.i.d. $g(U^{(l)})$ $(l = 1, \cdots, L)$. The new (unbiased) estimator of $\int_{[0,1]^s} g(x)dx$ is then

$$\frac{1}{I}\sum_{i=1}^{I} Z^{(i)} = \frac{1}{IL}\sum_{i=1}^{I}\sum_{l=1}^{L} g(\{U^{(i)} + \xi^{(l)}\}). \tag{2}$$

Using the Central Limit Theorem on the I independent random variables $Z^{(i)}$, $i = 1, \cdots, I$, we obtain a confidence interval for the integral. As shown by the following Theorem, the low discrepancy sequence allows to reduce the variance of the estimator:

Theorem 1 *(Tuffin 1996a,1997a) For every low discrepancy sequence* $\mathcal{P} = (\xi^{(l)})_{l\in\mathbb{N}}$ *over* $[0,1)^s$ *and every function* g *of bounded variation* $V(g)$*, then*

$$\sigma^2\left(\frac{1}{L}\sum_{l=1}^{L} g(\{U + \xi^{(l)}\})\right) = O(L^{-2}(\log L)^{2s}). \tag{3}$$

In the same way, if g is a Riemann-integrable function (or such that g and g^2 have a convergent generalized Riemann integral) such that there exist a step function h and a constant M with

$$\sup_{l\geq 1}\left|\frac{\sigma^2(h,L,\xi)-\sigma^2(g,L,\xi)}{\sigma^2(h,L,\xi)}\right|\leq M, \tag{4}$$

where $\sigma^2(f,L,\xi)=\sigma^2\left(\frac{1}{L}\sum_{l=1}^{L} f(\{\xi^{(l)}+U\})\right)$ for every function f, then (3) is satisfied as well.

The aggregate convergence speed of the estimator given by (2) is then in $O(I^{-1/2}L^{-1}(\ln L)^s)$ and can be faster for smooth functions (Tuffin 1998). In (Tuffin 1996a) we have found by means of numerical tests, that Sobol' sequences using the Gray code as implemented in (Press et al 1992) are the low discrepancy sequences which give the best results.

4.3 Application to loss networks problem

To use the randomized quasi-Monte Carlo method for the computation of (1), we may just consider now the function $g:[0,1]^K\to\mathbb{R}$ defined by

$$g(u_1,\cdots,u_K)=\sum_{n_1=0}^{N_1}\cdots\sum_{n_K=0}^{N_K}\Phi(n_1,\cdots n_K)\prod_{k=1}^{K}\mathbb{1}(F_k(n_k-1)<u_k\leq F_k(n_k)), \tag{5}$$

where $F_k(n)=\mathbb{P}(Y_k\leq n)$ is the distribution function of Y_k for $k=1,\cdots,K$. It can easily be seen that the random variable $g(U)$ with U uniformly distributed on $[0,1)^s$ has the same distribution function than $\Phi(\boldsymbol{Y})$ (as defined in Section 3).

However, this algorithm generates each random variable Y_k $(k=1,\cdots,K)$ in a time increasing with N_k, because we have to compare $u_k\in[0,1)$ with the values $F_k(n_k)$, $(n_k=0,\cdots,N_k)$. This can be improved using the alias algorithm (see (Fishman 1997) for a description) which generates Y_k $(k=1,\cdots,K)$ in a $O(1)$ time, that is, independent of N_k. We now briefly describe the function associated with this technique to be able to later find the convergence speed of the randomized quasi-Monte Carlo method. The alias algorithm uses

pre-computed tables Q_k, D_k and E_k $(k = 1, \cdots, K)$ to obtain the desired probability distribution (Fishman 1997). Define then

$$\mathbb{1}^{(d)}(u_k, n_k) = \mathbb{1}(\{(N_k+1)u_k\} \leq Q_k(\lfloor (N_k+1)u_k \rfloor), D_k(\lfloor (N_k+1)u_k \rfloor) = n_k)$$

and

$$\mathbb{1}^{(e)}(u_k, n_k) = \mathbb{1}(\{(N_k+1)u_k\} > Q_k(\lfloor (N_k+1)u_k \rfloor), E_k(\lfloor (N_k+1)u_k \rfloor) = n_k)$$

where $\lfloor u \rfloor$ is the largest integer less than or equal to $u \in \mathbb{R}$. The function g corresponding to this algorithm is then given by

$$g(u_1, \cdots, u_K) = \sum_{n_1=0}^{N_1} \cdots \sum_{n_K=0}^{N_K} \Phi(n_1, \cdots n_K) \times \prod_{k=1}^{K} \left[\mathbb{1}^{(d)}(u_k, n_k) + \mathbb{1}^{(e)}(u_k, n_k) \right]. \tag{6}$$

The variance of the randomized quasi-Monte Carlo estimator has the following convergence property:

Theorem 2 *For each choice of function g ((5) or (6)), the variance of the random variable*

$$\frac{1}{L} \sum_{l=1}^{L} g(\{U + \xi^{(l)}\})$$

is $O(L^{-2}(\ln L)^{2K})$.

Proof: the function g is a finite sum of indicator functions of rectangulars with sides parallel to the coordinate axis. Such a function is of bounded variation (Morokoff and Caflisch 1995; p. 226), so using Theorem 1 we obtain the result. □

5 Numerical illustrations

We apply the randomized quasi-Monte Carlo method to a network consisting in a star topology (on which usual algorithms can not be applied, (Ross 1995,page 240)) with four links as described in Figure 1. We take here the link capacities $C_1 = 20$, $C_2 = 30$, $C_3 = 40$ and $C_4 = 50$. There is traffic only between the

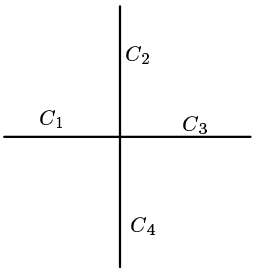

Figure 1: Star topology with four links

Class k	Route	b_k	ρ_k	γ_k
1	1,2	1	5.0	5.48
2	1,3	1	5.0	5.48
3	1,4	1	5.0	5.48
4	2,3	1	5.0	5.48
5	2,4	1	5.0	5.48
6	3,4	1	5.0	5.48
7	1,2	5	1.0	1.58
8	1,3	5	1.0	1.58
9	1,4	5	1.0	1.58
10	2,3	5	1.0	1.58
11	2,4	5	1.0	1.58
12	3,4	5	1.0	1.58

Table 1: Data of the star topology example with four links

six pairs of leaf nodes. They are two classes of traffic, then $K = 12$ classes of calls. The other specifications are in Table 1.

The results for Monte Carlo methods are given in Table 2, and for randomized quasi-Monte Carlo ones in Table 3. To reduce the size of each table, we only give the estimated confidence interval for loss probabilities of the two first classes of calls. As it can be observed in these Tables, considering the

I	Class	B_k estimation	CI Width
10^6	1	2.089900e-01	1.523e-02
10^6	2	1.763586e-01	1.436e-02
10^8	1	2.087760e-01	1.530e-03
10^8	2	1.752015e-01	1.439e-03
$2\ 10^9$	1	2.086761e-01	3.412e-04
$2\ 10^9$	2	1.750566e-01	3.210e-04

Table 2: Some Monte Carlo results for the star topology example with four links

I	L	Class	B_k estimation	CI Width
10^2	10^4	1	2.128194e-01	1.495e-02
10^2	10^4	2	1.811122e-01	1.465e-02
10^3	10^3	1	2.104176e-01	1.518e-02
10^3	10^3	2	1.748653e-01	1.445e-02
10^2	10^6	1	2.083600e-01	1.466e-03
10^2	10^6	2	1.745644e-01	1.382e-03
10^2	$2\ 10^7$	1	2.086721e-01	2.591e-04
10^2	$2\ 10^7$	2	1.750282e-01	2.516e-04

Table 3: Some randomized quasi-Monte Carlo results for the star topology example with four links

confidence interval widths, the improvements are not as large as expected in Theorem 2 using a moderate number L of points of the low discrepancy sequence (Sobol' one), but they begin to be significant with a quite large L.

We consider also a smaller example with a star topology but with only three links of respective capacities $C_1 = 20$, $C_2 = 30$ and $C_3 = 40$. Table 4 gives the other specifications regarding this example.

Class k	Route	b_k	ρ_k	γ_k
1	1,2	1	5.0	5.48
2	1,3	1	5.0	5.48
3	2,3	1	5.0	5.48
4	1,2	5	1.0	1.58
5	1,3	5	1.0	1.58
6	2,3	5	1.0	1.58

Table 4: Data of the star topology example with three links

The results for Monte Carlo simulation are given in Table 5, and for randomized quasi-Monte Carlo in Table 6. The improvement with low discrepancy

I	Class	B_k estimation	CI Width
10^6	1	9.471070e-02	2.081e-03
10^6	2	8.641567e-02	2.025e-03
10^8	1	9.478580e-02	2.083e-04
10^8	2	8.646147e-02	2.026e-04
10^9	1	9.472626e-02	6.586e-05
10^9	2	8.642010e-02	6.407e-05

Table 5: Some Monte Carlo results for the star topology example with three links

sequences is greater and more quickly significant in this case than for the example with four links.

The worse improvements (in confidence interval width) on the first example are due to the shape of the function to integrate: it consists in a larger collection of rectangular characteristic functions, and the good distribution of low discrepancy sequences needs in this case more time to be effective. More exactly, the greater the number of rectangular characteristic functions the

integrand is composed of, the longer is the time needed to notice the good approximation. However, for a sufficiently large number of points, the accuracy given in Theorem 2 can be found, as seen in Tables 2 and 3. At least, for a small number of used points of the low discrepancy sequence, the results when using randomized quasi-Monte Carlo method are not worse than when using Monte Carlo; moreover, for a large number of points, the improvement is significant, so that it is interesting to use a quite large L.

I	L	Class	B_k estimation	CI Width
10^2	10^4	1	9.443673e-02	1.569e-03
10^2	10^4	2	8.618917e-02	1.474e-03
10^4	10^2	1	9.386954e-02	1.992e-03
10^4	10^2	2	8.534427e-02	1.940e-03
10^2	10^6	1	9.474012e-02	8.841e-05
10^2	10^6	2	8.642337e-02	8.446e-05
10^2	10^7	1	9.473390e-02	1.746e-05
10^2	10^7	2	8.641809e-02	1.574e-05

Table 6: Some randomized quasi-Monte Carlo results for the star topology example with three links

Another important point to observe when studying the efficiency of a Monte Carlo estimator is the computational time (recall that the efficiency is $1/(\sigma_I^2 t)$ where σ_I^2 is the estimator's variance, with I random variables, and t its computational time). The ratios of computational times of the randomized quasi-Monte Carlo method using I random variables and L points of the Sobol' sequence with respect to the ones of Monte Carlo method using IL random variables are given in Table 7 for star topologies with three and four links. As it can be seen in Table 7, the gain in computational time is significant. Therefore, using the randomized quasi-Monte Carlo method is interesting: it is faster and gives smaller confidence interval widths when using many points.

Example	I	L	ratio
star3	100	10000	0.725
star3	10000	100	0.73
star3	100	10^6	0.71
star3	100	10^7	0.71
star4	100	10000	0.66
star4	1000	1000	0.66
star4	100	10^6	0.65
star4	100	$2\ 10^7$	0.64

Table 7: Ratio of computational times for the star topologies examples: time for the randomized quasi-Monte Carlo method using I random variables and L points of the Sobol' sequence divided by the time for Monte Carlo method using IL random variables

6 Direction for further work: the integral representation

In (Ross and Wang 1997), the Monte Carlo simulation using an integral representation of the normalization constant available for a confined class of product-form multi-class queuing networks is showed to be more efficient than the one using its summation form. In (Tuffin 1997b), the improvements when using randomized quasi-Monte Carlo methods are seen to be the largest on the integral representation. Therefore, a promising way to improve the previous results of this paper is to use (Kobayashi and Mark, 1997; equation (6.34)) where, with our current notations, the normalization constant G is expressed as

$$G = \frac{1}{(2\pi i)^J} \oint_\Gamma \frac{G^*(\boldsymbol{z})}{\boldsymbol{z}^{\mathbf{1}+\boldsymbol{C}}} d\boldsymbol{z} \tag{7}$$

where $\boldsymbol{z} = (z_1, \cdots, z_J) \in \mathbb{C}^J$, $\mathbf{1} = (1, \cdots, 1)$, $\boldsymbol{C} = (C_1, \cdots, C_J) \in \mathbb{N}^J$, $\boldsymbol{b}_k = (b_k \mathbb{1}(k \in \mathcal{K}_1), \cdots, b_k \mathbb{1}(k \in \mathcal{K}_J)) \in \mathbb{N}^J$, $\boldsymbol{z}^{\boldsymbol{b}_k} = \prod_{j=1}^J z_j^{b_j}$,

$$G^*(\boldsymbol{z}) = \prod_{j=1}^J \left(\frac{1}{1-z_j}\right) \exp\left(\sum_{k=1}^K \rho_k \boldsymbol{z}^{\boldsymbol{b}_k}\right),$$

and where Γ is a simple closed contour about the origin, i.e., a Cartesian product of, for each coordinate $z_i \in \mathbb{C}$ of $\boldsymbol{z}$, closed contours of 0 not including 1.

Note that this integral form is quite different of the one for product-form multi-class queuing networks because it is an integral over $\mathbb{C}^J$.

If a satisfactory Monte Carlo method is available, then turning it into a randomized quasi-Monte Carlo one may give an accuracy improved of several orders of magnitude because the dimension is J (often smaller than K) and because quasi-Monte Carlo methods are known to be faster on continuous functions (Tuffin 1997b).

However, we have not succeeded in finding a good Monte Carlo method to simulate the real part of (7), even when using the imaginary part as a control variate. The problem is that for quite large networks, we obtain very bad estimations whatever the choice of the path around the origin we have taken for each z_i (the exponent of the denominator in (7) is large and $|z_i| < 1$ at least near 1). Thus finding a good way to choose the paths is a challenging open problem.

References

R. Cranley and T.N.L. Patterson. 1976. Randomization of number theoretic methods for multiple integration. *SIAM J. Numer. Anal.*, 13:904–914.

G.S. Fishman. 1997. *Monte Carlo: Concepts, Algorithms and Applications.* Springer-Verlag.

P. J. Fleming, D. Schaeffer, and B. Simon. 1995. Efficient Monte-Carlo Simulation of a Product-Form Model for a Cellular System with Dynamic Resource Sharing. *ACM ToMaCS*, 5:3–21.

H. Kobayashi and B.L. Mark. 1997. Product-form loss networks. In J.H. Dshalalow, editor, *Frontiers in Queuing Systems: Models and Applications in Science and Engineering*, CRC Press, 147–195.

W. J. Morokoff and R. E. Caflisch. 1995. Quasi-Monte Carlo Integration. *Journal. Comput. Phys.*, 122:218–230.

H. Niederreiter. 1992. *Random Number Generation and Quasi-Monte Carlo Methods.* CBMS-SIAM 63, Philadelphia.

W.H. Press, S.A. Teutolsky, W.T. Vetterling, and B.P. Flannery. 1992. *Numerical recipes in C: the art of scientific computing.* Cambridge University Press.

K. W. Ross and J. Wang. 1992. Monte carlo summation applied to product-form loss networks. *Prob. Eng. Info. Sci.*: 323–348.

K.W. Ross. 1995. *Multiservice Loss Models for Broadband Telecommunication Networks.* Springer-Verlag.

K.W. Ross and J. Wang. 1997. Implementation of Monte Carlo Integration for the Analysis of product-form queueing networks. *Perf. Eval.*, 29:273–292.

B. Tuffin. 1996a. On the Use of Low Discrepancy Sequences in Monte Carlo Methods. *Monte Carlo Methods and Applications*, 2:295–320, and 4:87–90. 1998

B. Tuffin. 1996b. Variance reduction technique for a cellular system with dynamic resource sharing. In *Proceedings of the 10th European Simulation Multiconference*, pages 467–471.

B. Tuffin. 1997a. *Simulation accélérée par les méthodes de Monte Carlo et quasi-Monte Carlo : théorie et applications.* PhD thesis, Université de Rennes 1.

B. Tuffin. 1997b. Variance Reductions applied to Product-Form Multi-Class Queuing Network. *ACM ToMaCS*, 7:478–500.

B. Tuffin. 1998. Variance reduction order using good lattice points in Monte Carlo methods. *Computing*, 61:371–378.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399